



21st CENTURY
案例型规划教材

21世纪全国高职高专计算机案例型规划教材

C语言程序 设计经典案例教程

主 编 韦良芬 王 勇
主 审 张佑生



北京大学出版社
PEKING UNIVERSITY PRESS



《CAXA 软件应用案例教程》

主编 王 强
副主编 王 强

内 容 简 介

21 世纪全国高职高专计算机案例型规划教材

C 语言程序设计经典案例教程

主 编	韦良芬	王 勇			
副主编	张继山	张 健	余久久	咎 朝	
	张 迪	韩高峰	高 凤		
参 编	乔 忠	余会娟			
主 审	张佑生				



北京大学出版社
PEKING UNIVERSITY PRESS

定价：28.00 元

内 容 简 介

本书系统地介绍 C 语言基础知识及相关概念、结构化程序设计方法及 C 程序开发环境的应用。本书共分为 10 章, 主要内容包括: C 语言概述、C 语言基础、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、结构与枚举类型、预处理与文件、结构化程序设计案例。

本书以初学者对 C 语言的认识和“超市管理系统”作为主线, 设置学习情境。在结构安排上先设置情境提出问题, 让读者带着问题去学习; 在内容组织上, 基本遵循理论知识够用的原则, 注重编程思想和解决实际问题的能力培养。

本书既可以作为计算机及相关专业的教材, 也可作为自学教材以及各类计算机等级考试的培训教材。

图书在版编目(CIP)数据

C 语言程序设计经典案例教程/韦良芬, 王勇主编. —北京: 北京大学出版社, 2010.6

(21 世纪全国高职高专计算机案例型规划教材)

ISBN 978-7-301-17337-4

I. ①C… II. ①韦…②王… III. ①C 语言—程序设计—高等学校: 技术学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 107362 号

书 名: C 语言程序设计经典案例教程

著作责任者: 韦良芬 王 勇 主编

策 划 编 辑: 李彦红

责 任 编 辑: 刘国明

标 准 书 号: ISBN 978-7-301-17337-4/TP · 1111

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> <http://www.pup6.com>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电 子 邮 箱: pup_6@163.com

印 刷 者: 河北滦县鑫华书刊印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787mm×1092mm 16 开本 15.75 印张 354 千字

2010 年 6 月第 1 版 2010 年 6 月第 1 次印刷

定 价: 28.00 元

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有 侵权必究

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

21 世纪全国高职高专计算机案例型规划教材

专家编写指导委员会

主 任	刘瑞挺	南开大学
副主任	安志远	北华航天工业学院
	丁桂芝	天津职业大学
委 员	(按拼音顺序排名)	
	陈 平	马鞍山师范高等专科学校
	褚建立	邢台职业技术学院
	付忠勇	北京政法职业技术学院
	高爱国	淄博职业学院
	黄金波	辽宁工程技术大学职业技术学院
	李 缨	中华女子学院山东分院
	李文华	湖北仙桃职业技术学院
	李英兰	西北大学软件职业技术学院
	田启明	温州职业技术学院
	王成端	潍坊学院
	王风华	唐山工业职业技术学院
	薛铁鹰	北京农业职业技术学院
	张怀中	湖北职业技术学院
	张秀玉	福建信息职业技术学院
	赵俊生	甘肃省合作民族师范高等专科学校
顾 问	马 力	微软(中国)公司 Office 软件资深教师
	王立军	教育部教育管理信息中心

信息技术的案例型教材建设

(代丛书序)

刘瑞挺/文

北京大学出版社第六事业部在 2005 年组织编写了两套计算机教材,一套是《21 世纪全国高职高专计算机系列实用规划教材》,截至 2008 年 6 月已经出版了 80 多种;另一套是《21 世纪全国应用型本科计算机系列实用规划教材》,至今已出版了 50 多种。这些教材出版后,在全国高校引起热烈反响,可谓初战告捷。这使北京大学出版社的计算机教材市场规模迅速扩大,编辑队伍茁壮成长,经济效益明显增强,与各类高校师生的关系更加密切。

2007 年 10 月北京大学出版社第六事业部在北京召开了“21 世纪全国高职高专计算机案例型教材建设和教学研讨会”,2008 年 1 月又在北京召开了“21 世纪全国应用型本科计算机案例型教材建设和教学研讨会”。这两次会议为编写案例型教材做了深入的探讨和具体的部署,制定了详细的编写目的、丛书特色、内容要求和风格规范。在内容上强调面向应用、能力驱动、精选案例、严把质量;在风格上力求文字精练、脉络清晰、图表明快、版式新颖。这两次会议吹响了提高教材质量第二战役的进军号。

案例型教材真能提高教学的质量吗?

是的。著名法国哲学家、数学家勒内·笛卡儿(Rene Descartes, 1596~1650)说得好:“由一个例子的考察,我们可以抽出一条规律。”(From the consideration of an example we can form a rule.)事实上,他发明的直角坐标系,正是通过生活实例得到的灵感。据说是在 1619 年夏天,笛卡儿因病住进医院。中午他躺在病床上苦苦思索一个数学问题时,忽然看到天花板上有一只苍蝇飞来飞去。当时天花板是用木条做成正方形的格子。笛卡儿发现,要说出这只苍蝇在天花板上的位置,只需说出苍蝇在天花板上的第几行和第几列。当苍蝇落在第四行、第五列的那个正方形时,可以用(4, 5)来表示这个位置……由此他联想到可用类似的办法来描述一个点在平面上的位置。他高兴地跳下床,喊着“我找到了,找到了”,然而不小心把国际象棋撒了一地。当他的目光落到棋盘上时,又兴奋地一拍大腿:“对,对,就是这个图”。笛卡儿锲而不舍的毅力,苦思冥想的钻研,使他开创了解析几何的新纪元。千百年来,代数与几何井水不犯河水。17 世纪后,数学突飞猛进的发展,在很大程度上归功于笛卡儿坐标系和解析几何学的创立。

这个故事,听起来与阿基米德在浴池洗澡而发现浮力原理,牛顿在苹果树下遇到苹果落到头上而发现万有引力定律,确有异曲同工之妙。这就证明,一个好的例子往往能激发灵感,由特殊到一般,联想出普遍的规律,即所谓的“一叶知秋”、“见微知著”的意思。

回顾计算机发明的历史,每一台机器、每一颗芯片、每一种操作系统、每一类编程语言、每一个算法、每一套软件、每一款外部设备,无不像闪光的珍珠串在一起。每个案例都闪烁着智慧的火花,是创新思想不竭的源泉。在计算机科学技术领域,这样的案例就像大海岸边的贝壳,俯拾皆是。

事实上,案例研究(Case Study)是现代科学广泛使用的一种方法。Case 包含的意义很广,包括 Example 例子,Instance 事例、示例,Actual State 实际状况, Circumstance 情况、事件、境遇,甚至 Project 项目、工程等。

大家知道在计算机的科学术语中,很多是直接来自日常生活的。例如 Computer 一词早在 1646 年就出现于古代英文字典中,但当时它的意义不是“计算机”而是“计算工人”,即专门从事简单计算的工人。同样的,Printer 的意义当时也是“印刷工人”而不是“打印机”。正是由于这些“计算工人”和“印刷工人”常出现计算错误和印刷错误,才激发查尔斯·巴贝奇(Charles Babbage, 1791—1871)设计了差分机和分析机,这是最早的专用计算机和通用计算机。这位英国剑桥大学数学教授、机械设计专家、经济学家和哲学家是国际公认的“计算机之父”。

20 世纪 40 年代,人们还用 Calculator 表示计算机。到电子计算机出现后,才用 Computer 表示计算机。此外,硬件(Hardware)和软件(Software)来自销售人员,总线(Bus)就是公共汽车或大巴,故障和排除故障源自格瑞斯·霍普(Grace Hopper, 1906—1992)发现的“飞蛾子”(Bug)和“抓蛾子”或“抓虫子”(Debug)。其他如鼠标、菜单……不胜枚举。至于哲学家进餐问题、理发师睡觉问题更是操作系统文化中脍炙人口的经典。

以计算机为核心的信息技术,从一开始就与应用紧密结合。例如,ENIAC 用于弹道曲线的计算,ARPANET 用于资源共享以及核战争时的可靠通信。即使是非常抽象的图灵机模型,也受到“二战”时图灵博士破译纳粹密码工作的影响。

在信息技术中,既有许多成功的案例,也有不少失败的案例;既有先成功而后失败的案例,也有先失败而后成功的案例。好好研究它们的成功经验和失败教训,对于编写案例型教材有重要的意义。

我国正在实现中华民族的伟大复兴,教育是民族振兴的基石。改革开放 30 年来,我国高等教育在数量上、规模上已有相当大的发展。当前的重要任务是提高培养人才的质量,为此,培养模式必须从学科知识的灌输转变为素质与能力的培养。应当指出,大学课堂在高新技术的武装下,利用 PPT 进行的“高速灌输”、“翻页宣科”有愈演愈烈的趋势,我们不能容忍用“技术”绑架教学,而是让教学工作乘信息技术的东风自由地飞翔。

本系列教材的编写,以学生就业所需的专业知识和操作技能为着眼点,在适度的基础知识与理论体系覆盖下,突出应用型、技能型教学的实用性和可操作性,强化案例教学。本套教材将会融入大量最新的示例、实例以及操作性较强的案例,力求提高教材的趣味性和实用性,打破传统教材自身知识框架的封闭性,强化实际操作训练,使本系列教材做到“教师易教,学生乐学,技能实用”。有了广阔的应用背景,再造计算机案例型教材就有了基础。

我相信北京大学出版社在全国各地高校教师的积极支持下,精心设计,严格把关,一定能够建设出一批符合计算机应用型人才培养模式的、以案例型为创新点和兴奋点的精品教材,并且通过一体化设计实现多种媒体有机结合的立体化教材,为各门计算机课程配齐电子教案、学习指导、习题解答、课程设计等辅导资料。让我们用锲而不舍的毅力,勤奋好学的钻研,向着共同的目标努力吧!

刘瑞挺教授 本系列教材编写指导委员会主任、全国高等院校计算机基础教育研究会副会长、中国计算机学会普及工作委员会顾问、教育部考试中心全国计算机应用技术证书考试委员会副主任、全国计算机等级考试顾问。曾任教育部理科计算机科学教学指导委员会委员、中国计算机学会教育培训委员会副主任。PC Magazine《个人电脑》总编辑、CHIP《新电脑》总顾问、清华大学《计算机教育》总策划。

前 言

随着信息技术的发展,计算机在人类的日常生活中扮演着越来越重要的角色。用计算机语言编写小型应用软件,进而较深入地学习和运用计算机知识驾驭计算机是计算机相关专业人员的重要技能之一。而 C 语言以其短小精悍、使用灵活、功能强大等特点深受人们的喜爱。

对一般初学者来说, C 语言规则较多,使用太灵活,不易掌握;而且 C 语言的应用范围越来越广,所涉及的知识也在不断增加,这也给学习增加了难度;另外,能够把计算机应用于解决实际生活问题也是现代人的一项重要技能要求。基于此,我们根据多年的教学经验,在传统教程的基础上进行了大胆的创新,以希望提高广大学习者的兴趣,激发广大学习者的主动性和创造性。

本书的主要特点如下。

(1) 本书以初学者的学习过程及其对“超市管理系统”的认识作为学习情境贯穿始终。一方面可以提高大家的学习兴趣;另一方面可以加深初学者对各章节内容在实际应用中的认识。

(2) 将 C 语言的精华——指针分散在各个相关章节中,以降低指针的学习难度,增加指针的辐射面积。

(3) 本书在结构安排和例题分析中,强调程序设计思想和方法的灌输,让学习者逐渐学会主动编程的方法。

(4) 本书在结构安排上还强调学生预习在前、教师教学在后,以引导学生的学习主动性。

(5) 本书例题新颖,在注重实际应用的基础上兼顾当前高等院校和社会上举办的各种计算机等级考试。

本书共分为 10 章,主要内容包括: C 语言概述, C 语言基础,顺序结构程序设计,循环结构程序设计,数组,函数,结构与枚举类型,预处理与文件,结构化程序设计案例。其中每个章节都安排具有丰富教学经验的老师负责编写完成。参加本书编写的有:韦良芬、王勇、张继山、张健、余久久、咎朝、张迪、韩高峰、高凤、乔忠、余会娟,其中韦良芬和王勇担任主编,张继山、张健、余久久、咎朝、张迪、韩高峰、高凤等担任副主编,博士生导师张佑生教授担任主审。另外,本书由韦良芬负责的组织、统筹与规划工作;由韦良芬、王勇和咎朝负责统稿、修改、审核和定稿工作。同时本书在编写的过程中受到了张佑生教授的悉心指导,也受到了安徽三联学院计算机科学与技术系领导和各位老师的大力支持,在此一并表示诚挚的谢意。

由于作者水平有限,书中难免会有疏漏和不足之处,恳请广大读者批评、指正,不胜感激。

编 者

2010 年 3 月

目 录

第1章 C语言概述.....1

1.1 各种计算机语言简介.....2

1.2 C语言的学习背景及其与其他

课程的联系.....3

1.3 简单C程序及其构成要素.....4

1.3.1 简单C程序实例.....4

1.3.2 C程序构成要素.....5

1.3.3 保留字和标识符.....6

1.4 C程序的运行环境.....7

1.4.1 C语言应用程序的处理流程.....7

1.4.2 Visual C++ 6.0 上机环境介绍.....8

本章小结.....14

下章引例.....14

习题.....14

第2章 C语言基础.....17

2.1 C语言的基本数据类型.....18

2.2 常量和变量.....18

2.2.1 常量与符号常量.....18

2.2.2 变量的定义.....21

2.2.3 变量的要素.....22

2.2.4 变量的类型转换.....23

2.3 指针变量的概念和定义.....25

2.3.1 地址和指针的概念.....25

2.3.2 指针变量的定义.....26

2.3.3 指针变量的引用.....26

2.4 运算符与表达式.....28

2.4.1 算术运算符与算术表达式.....28

2.4.2 赋值运算符与赋值表达式.....31

2.4.3 逗号运算符与逗号表达式.....31

2.4.4 求字节数运算符 sizeof.....32

2.4.5 位运算符及其表达式.....32

2.4.6 运算符的优先级与结合性.....33

2.5 综合实训.....33

录

综合实训一.....33

综合实训二.....34

2.6 情境案例分析.....34

本章小结.....35

下章引例.....36

习题.....36

第3章 顺序结构程序设计.....39

3.1 输入/输出函数.....40

3.1.1 字符数据的输入/输出.....40

3.1.2 格式输入与输出函数.....41

3.2 软件开发过程.....48

3.2.1 程序设计的一般流程.....48

3.2.2 软件开发流程.....50

3.3 结构化程序设计的基本概念.....51

3.3.1 程序的3种基本结构.....51

3.3.2 C语言的语句.....51

3.4 顺序结构程序设计案例分析.....53

3.5 综合实训.....54

3.6 情境案例分析.....55

本章小结.....56

下章引例.....57

习题.....57

第4章 选择结构程序设计.....60

4.1 选择结构问题基本解决步骤和 方法.....61

4.2 条件的描述.....62

4.2.1 关系运算符和关系表达式.....63

4.2.2 逻辑运算符和逻辑表达式.....63

4.3 if语句.....65

4.3.1 if语句的3种形式.....65

4.3.2 if语句的嵌套.....68

4.4 if语句应用举例.....70

4.5 switch语句.....72

4.6 综合实训	74	6.3.3 二维数组元素的引用	116
综合实训一	74	6.3.4 二维数组应用举例	116
综合实训二	75	6.4 字符数组与字符串	118
4.7 情境案例分析	76	6.4.1 一维字符数组与字符串	118
本章小结	78	6.4.2 二维字符数组与字符串	120
下章引例	78	6.4.3 常用字符串处理函数	122
习题	78	6.4.4 字符数组应用举例	125
第 5 章 循环结构程序设计	83	6.5 指针数组	126
5.1 循环问题解决步骤和方法	84	6.5.1 指针数组的定义与初始化	126
5.2 while 语句	85	6.5.2 指针数组元素的引用	127
5.3 do...while 语句	86	6.6 指针与数组	128
5.4 for 语句	88	6.6.1 指向数组元素的指针	128
5.5 转移语句	90	6.6.2 通过指针引用数组元素	129
5.5.1 goto 语句	91	6.6.3 指向一维数组的指针变量	130
5.5.2 break 语句	91	6.6.4 指向二维数组的指针变量	131
5.5.3 continue 语句	92	6.6.5 字符指针与字符串	134
5.6 for、while 和 do...while 三种 循环语句的比较	93	6.7 综合实训	136
5.7 循环的嵌套	93	综合实训一	136
5.8 综合实训	96	综合实训二	137
综合实训一	96	综合实训三	138
综合实训二	97	6.8 情境案例分析	139
综合实训三	98	本章小结	141
综合实训四	99	下章引例	141
5.9 情境案例分析	100	习题	142
本章小结	101	第 7 章 函数	145
下章引例	101	7.1 模块化编程的基本方法和步骤	146
习题	102	7.1.1 模块化编程方法	146
第 6 章 数组	106	7.1.2 模块化编程步骤	146
6.1 解决数组问题的基本方法和步骤	107	7.2 函数的定义和调用	146
6.2 一维数组	107	7.2.1 函数的定义	146
6.2.1 一维数组的定义	107	7.2.2 函数的调用	148
6.2.2 一维数组的初始化	109	7.3 函数的嵌套调用	151
6.2.3 一维数组元素的引用	110	7.4 函数的递归调用	153
6.2.4 一维数组应用举例	111	7.5 数组作为函数参数	154
6.3 二维数组	113	7.5.1 数组元素作为函数实参	154
6.3.1 二维数组的定义	114	7.5.2 数组名作为函数参数	155
6.3.2 二维数组的初始化	115	7.6 函数的指针和指向函数的 指针变量	156
		7.6.1 指向函数的指针变量	156

7.6.2 函数指针变量定义及调用.....156	9.1.1 宏定义.....202
7.6.3 指针变量作为函数的参数.....157	9.1.2 文件包含.....204
7.7 变量的作用域和存储类别.....160	9.1.3 条件编译.....205
7.7.1 变量的作用域.....160	9.2 文件.....206
7.7.2 变量的存储类别.....162	9.2.1 文件概述.....206
7.8 综合实训.....165	9.2.2 文件指针.....207
综合实训一.....165	9.2.3 文件操作.....207
综合实训二.....166	9.3 情境案例分析.....213
7.9 情境案例分析.....168	本章小结.....215
本章小结.....171	下章引例.....215
下章引例.....171	习题.....216
习题.....171	
第8章 结构与枚举类型.....175	第10章 结构化程序设计案例.....219
8.1 结构体类型.....176	10.1 结构化设计流程概述.....220
8.1.1 结构体类型的定义.....176	10.2 问题分析.....220
8.1.2 结构体类型变量的定义.....177	10.2.1 需求分析.....221
8.1.3 结构体变量的初始化.....179	10.2.2 需求规格说明书.....221
8.1.4 访问结构体变量成员的方法.....180	10.3 设计.....223
8.1.5 结构体数组.....182	10.3.1 模块.....223
8.1.6 指向结构体类型数据的指针.....185	10.3.2 程序设计准则.....224
8.2 枚举类型.....189	10.3.3 “键盘练习”程序总体设计分析.....225
8.2.1 枚举类型定义.....189	10.4 实现.....226
8.2.2 枚举类型变量的定义.....190	10.4.1 编码.....226
8.2.3 枚举变量的应用.....191	10.4.2 “键盘练习”程序源代码.....226
8.3 类型定义.....192	10.5 测试.....229
8.4 综合实训.....194	10.5.1 测试基本知识.....229
8.5 情境案例分析.....195	10.5.2 “键盘打字程序”功能测试过程简介.....230
本章小结.....196	本章小结.....232
下章引例.....197	习题.....232
习题.....197	
第9章 预处理与文件.....201	附录A.....233
9.1 预处理.....202	附录B.....234
	参考文献.....235

第1章

C语言概述



学习情境

张明同学是大学一年级计算机及应用专业的一名学生，他对程序设计充满着好奇，对软件开发人员更是敬佩有加，可是他没有基础，不知道从哪学起，也不知该怎么去学，于是他去咨询老师：怎样才能学好编程技术？首先该从什么语言学起呢？老师告诉他：要学习编程，首先要有恒心，切忌浮躁，慢慢学习，多动手；对于没有程序设计基础的入门人员，首先应该从C语言程序设计开始学起，所有的语言都是相通的，学会了C语言的编程思想，其他的编程语言都会变得非常简单；当然，在学习C语言之前了解一下各种语言的种类和特点还是很有必要的。



预习要点及目标

- 了解计算机语言的发展及特点；
- 了解C语言的相关特点及学习C语言的必要性；
- 掌握C程序的基本结构；
- 掌握标识符的命名规则；
- 掌握C程序的运行环境及上机过程。



情境知识关联

人类可以通过各种软件控制计算机硬件，从而使计算机能够按照要求处理并解决日常生活中的各种问题，而语言是一种实现人类处理问题的过程和思想的重要手段。C语言是国际上广泛流行的计算机高级程序设计语言，它集高级语言和低级语言的功能于一体，除了可用于操作系统的开发，也适合于工业控制、智能仪表、嵌入式系统和硬件驱动等方向。

1.1 各种计算机语言简介

当前指令流的计算机是以“程序存储，程序执行”为基本原理。要使计算机能够完成特定的工作任务，就必须编制相应的程序。编制程序，就需要使用相应的程序设计语言。计算机程序设计语言，经历了从机器语言、汇编语言到高级语言的发展历程。

1. 机器语言

电子计算机所使用的是由“0”和“1”组成的二进制数，二进制是计算机语言的基础。计算机发明之初，人们只能去适应计算机的要求，用计算机的语言去命令计算机处理各种问题，也就是写出一串串由“0”和“1”组成的指令序列交由计算机执行。这种语言就是机器语言。使用机器语言是十分复杂的，特别是在程序有错需要修改时，更是如此。而且，由于每种系列的计算机指令系统往往各不相同，所以在一台计算机上执行的程序，要想在另一台计算机上执行，可能得重新编写程序，从而造成了工作的重复。但由于使用的是针对特定型号计算机的语言，故而运算效率是所有语言中最高的。机器语言是第一代计算机语言。

2. 汇编语言

为了减小使用机器语言编程的难度，人们进行了一种有益的改进：用一些简洁的英文字母、符号串来替代特定的指令的二进制串。比如，用“ADD”代表加法，“MOV”代表数据传递等。这样人们就很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言就称为汇编语言，即第二代计算机语言。然而计算机是不认识这些符号的，这就需要有一个专门的程序，专门负责将这些符号翻译成二进制数的机器语言，这种翻译程序被称为汇编程序。

汇编语言同样十分依赖于机器硬件，移植性不好，但效率仍十分高，针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精练而质量高，所以至今仍是一种常用而强有力的软件开发工具。

3. 高级语言

从最初与计算机交流的困难经历中，人们意识到，应该设计一种这样的语言，这种语言接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用。经过努力，1954年，第一个完全脱离机器硬件的高级语言——FORTRAN问世了。40多年来，共有几百种高级语言出现，其中有重要意义的有几十种，影响较大、使用较普遍的有FORTRAN、ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C、Prolog、Ada、C++、VC、VB、Delphi、Java等。

高级语言的发展也经历了从早期语言到结构化程序设计语言，从面向过程到非过程化程序语言的过程。相应的，软件的开发也由最初的个体手工作坊式的封闭式生产，发展为产业化、流水线式的工业化生产。

20世纪60年代中后期，软件越来越多，规模越来越大，而软件的生产基本上是人自为战，缺乏科学规范的系统规划与测试、评估标准，其恶果是大批耗费巨资建立起来的软件系统，由于含有错误而无法使用，甚至带来巨大损失，软件给人的感觉是越来越不可靠，以致几乎没有不出错的软件。这一切，极大地震动了计算机界，史称“软件危机”。人们认识到：大型程序的编制不同于写小程序，它应该是一项新的技术，应该像处理工程一样处理软件研制的全过程。

程序的设计应易于保证正确性,也便于验证正确性。1969年,提出了结构化程序设计方法,1970年,第一个结构化程序设计语言——Pascal语言出现,标志着结构化程序设计时期的开始。

20世纪80年代初开始,在软件设计思想上,又产生了一次革命,其成果就是面向对象的程序设计。在此之前的高级语言,几乎都是面向过程的,程序的执行是流水线式的,在一个模块被执行完成前,人们不能干别的事,也无法动态地改变程序的执行方向。这和人们日常处理事物的方式是不一致的,对人而言是希望发生一件事就处理一件事,也就是说,不能面向过程,而应是面向具体的应用功能,也就是对象(object)。其方法就是软件的集成化,如同硬件的集成电路一样,生产一些通用的、封装紧密的功能模块,称为软件集成块,它与具体应用无关,但能相互组合,完成具体的应用功能,同时又能重复使用。对使用者来说,只关心它的接口(输入量、输出量)及能实现的功能,至于如何实现的,那是它内部的事,使用者完全不用关心,C++、VB、Delphi就是典型代表。

高级语言的下一个发展目标是面向应用,也就是说:只需要告诉程序你要干什么,程序就能自动生成算法,自动进行处理。这就是非过程化的程序语言。

提示:高级语言的发展经历了从早期语言到结构化程序设计语言,从面向过程到非过程化程序语言的过程,高级语言的下一个发展目标是面向应用。

1.2 C语言的学习背景及其与其他课程的联系

1. C语言出现的历史背景

C语言是由美国贝尔实验室的D.M.Ritchie于1972年在B语言的基础上设计出来的。它既保持了B语言精炼和接近硬件的特点,又克服了B语言过于简单和数据无类型等不足。不过最初的C语言只是为描述和实现UNIX操作系统而提供的一种工具语言,并主要在贝尔实验室内部使用。1977年出现了不依赖于具体机器的C语言编译文本《可移植C语言编译程序》。C语言因此具有良好的可移植性,使UNIX操作系统迅速在各种机器上运行,并与此相辅相成,也使C语言得到了迅速推广和广泛应用。1978年以后,C语言已完全独立于UNIX,先后移植到大、中、小和微型计算机上,成为计算机程序设计语言的主流语种。

1978年,Brian W.Kernighan和Dennis M.Ritchie(合称K&R)正式出版了著名的*The C Programming Language*一书。此书中介绍的C语言成为后来广泛使用的C语言版本基础,它被称为标准C语言。

1983年,美国国家标准化协会根据问世以来的各种版本对C语言的发展和扩充,制定颁布了C语言的新标准ANSI C;1987年,美国国家标准化协会在综合各种C语言版本的基础上又颁布了新标准,称87 ANSI C;1990年,国际标准化组织ISO接受了87 ANSI C为ISO C的标准,这就是目前功能最完善,性能最好的C语言版本。

2. C语言的特点

C语言之所以发展迅速,成为最受欢迎的语言之一,主要因为它具有强大的功能。归纳起来,C语言主要具有以下特点。

(1) C语言既具有高级语言的特点,又具有低级语言的功能。

C语言能把高级语言的基本结构和语句与低级语言的实用性结合起来。一方面,C语言是

面向问题的程序设计语言，独立于计算机的硬件，对具体的算法进行描述，它的特点是独立性、通用性和可移植性。另一方面，C 语言可以像汇编语言一样对位、字节和地址进行操作，实现对计算机硬件的控制。另外 C 语言中还可以嵌入汇编子程序，使 C 语言具有更强大的底层控制能力。

(2) C 语言是结构化程序设计语言。

结构化程序设计语言的显著特点是代码及数据的分隔化，即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护及调试。C 语言是以函数形式提供给用户的，这些函数可方便地调用，并具有多种循环、条件语句控制程序流向，从而使程序完全结构化。

(3) C 语言数据结构丰富，功能齐全。

C 语言具有非常丰富的数据结构和类型，支持几乎所有数据的存储、计算和输出，并引入了指针的概念，可使程序效率更高。另外，C 语言也具有强大的图形功能，支持多种显示器和驱动器，而且计算功能、逻辑判断功能也比较强大，可以实现决策目的。

(4) C 语言适用范围广。

C 语言还有一个突出的优点就是适合多种操作系统，如 DOS、UNIX，也适用于多种机型。很多新型的语言都衍生自 C 语言，如 C++、Java 等。掌握了 C 语言，就掌握了很多门语言。通过 C 语言的学习，就可以更加方便地学习其他高级语言。

3. C 语言与其他课程的联系

在学习 C 语言程序设计之前，必须较为全面地了解数据在计算机中的表示与存储、计算机结构等方面的专业知识。本门课程的前修课程为“计算机文化基础”或“计算机导论”，其后继课程为“数据结构”、“算法基础”等。

1.3 简单 C 程序及其构成要素

用 C 语言语句编写的程序称为 C 语言源程序。C 程序是一种函数结构，一般由一个或若干个函数组成，这里通过几个简单实例介绍 C 语言程序的基本结构及其组成要素。

1.3.1 简单 C 程序实例

例 1-1 在显示器上显示语句“万丈高楼平地起！”。

```
#include "stdio.h"           /*文件包含*/
main()                       /*主函数*/
{                             /*一对大括号{}包含了程序的功能部分，即函数体*/
    printf("万丈高楼平地起！"); /*输出语句“万丈高楼平地起！”*/
}
```

例 1-2 计算两整数之和，并输出结果。

```
#include "stdio.h"
main()
{
    int a,b,sum;              /*定义变量，程序的声明部分*/
    printf("please input a and b:\n"); /*以下几行为程序的执行部分*/
}
```

```
scanf("%d,%d",&a,&b);
sum=a+b;
printf("sum=%d",sum);
}
```

例 1-3 求两整数中的较大值并输出。

```
#include "stdio.h"
main( )
{
    int a,b,c;
    scanf("%d,%d",&a,&b);          /*输入变量 a 和 b 的值*/
    c=max(a,b);                    /*调用 max 函数，将得到的值赋给 c*/
    printf("max=%d",c);            /*输出 c 的值*/
}
int max(int x,int y)               /*定义 max 函数，函数值为整型，形式参数 x,y 为整型*/
{
    int z;                         /*max 函数中的声明部分*/
    if(x>y) z=x;
    else z=y;
    return(z);                     /*将 z 的值返回，并带回调用处*/
}
```

1.3.2 C 程序构成要素

1. 注释

C 语言的注释是以“/*”开头并以“*/”结尾，在“/*”和“*/”之间的部分即为注释。程序编译时，不对注释作任何处理。注释可出现在程序中的任何位置，用来向用户提示或解释程序的意义，从而提高程序的可读性。在调试程序中对暂不使用的语句也可用注释符括起来，使编译跳过不作处理。

2. 文件包含指令#include

所谓“文件包含”是指一个源文件将另外一个源文件的全部内容包含进来，即将另外的文件包含到本文件之中，无需重新定义。被包含的文件称为头文件，常以.h 为后缀。C 语言提供了文件包含命令#include 来实现文件包含的操作。

#include 命令的一般形式为：

#include "文件名"或 #include <文件名>

如例 1-1 至例 1-3 都用到了代码：

```
#include "stdio.h"
```

3. main 函数

main 函数为主函数，是 C 程序执行的入口点。C 程序是一种函数结构，故一般由一个或若干个函数组成。从上面的例 1-1 至例 1-3 可以看出。

(1) C 程序是由函数构成的。

一个 C 源程序有且只有一个 main 函数，也可以包含一个 main 函数和若干个其他函数。因此，函数是 C 程序的基本单位。被调用的函数可以是系统提供的库函数(例如 scanf 函数和

printf 函数), 也可以是用户根据需要自己编写设计的函数(例如, 例 1-3 中的 max 函数)。

(2) 一个 C 程序总是从 main 函数开始执行, 而无论 main 函数在整个程序中的位置如何。

(3) main 函数的一般格式为:

```
main()
{
    函数体;
}
```

4. 语句

一个 C 语言程序是由一系列语句组成的。每一条语句都用来完成某一特定的操作任务。C 语言中每条语句的最后都用“;”标识。一个函数中的语句通常分为声明语句和执行语句。

1) 声明语句

声明语句通常用于定义变量、定义自定义类型、自定义函数说明、外部变量说明等, 如例 1-2 中的语句:

```
int a,b,sum;
```

就是声明语句, 用于声明变量。

2) 执行语句

执行语句是在程序执行时发出的某种动作, 用于完成程序题目要求的计算处理, 如例 1-2 中的语句:

```
printf("please input a and b:\n");
scanf("%d,%d",&a,&b);
sum=a+b;
printf("sum=%d",sum);
```

都为执行语句。

1.3.3 保留字和标识符

1. 保留字

保留字即关键字, 是 C 语言中已有的具有特定意义的字符符号。保留字不能用于其他目的。在 C 语言中有 32 个关键字, 所有关键字都是小写的。

32 个关键字如下:

auto default float register struct volatile break do for return switch while
case double goto short typedef char else if signed union const enum int sizeof
unsigned continue extern long static void

2. 标识符

计算机语言主要是由其所限定的符号表示一些字、词、命令、变量、数据等, 在 C 语言程序中, 为了区别各个变量、函数、标签和其他用户定义的对象, 都必须为它们取不同的名字, 这些名字称为标识符。

在 C 语言中定义标识符时, 必须遵守以下规则。

(1) 只能由英文字母、数字和下划线构成。